```
import pandas as pd
import matplotlib.pyplot as plt
```

```
data = pd.read_csv("/content/Food-Truck-LineReg.csv",names=['x','y'])
```

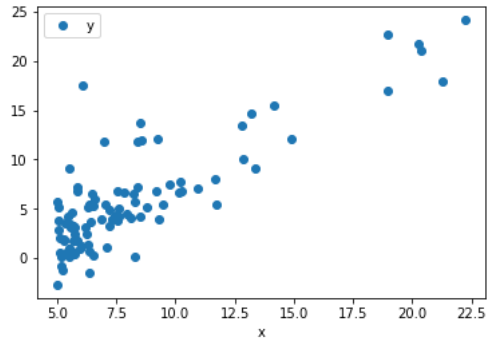
```
data
```

	x	y
0	6.1101	17.59200
1	5.5277	9.13020
2	8.5186	13.66200
3	7.0032	11.85400
4	5.8598	6.82330
...	...	...
92	5.8707	7.20290
93	5.3054	1.98690
94	8.2934	0.14454
95	13.3940	9.05510
96	5.4369	0.61705

97 rows × 2 columns

```
data.plot(x="x",y="y",style="o")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa9733f66d0>
```




```
# we need to find the mean of x and y
x_mean = data["x"].mean()
y_mean = data["y"].mean()
print(x_mean,y_mean)
```

```
8.159799999999999 5.839135051546393
```

```
# we need to have other things in the table
data["X"] = data["x"] - x_mean
data["Y"] = data["y"] - y_mean
data["X*Y"] = data["X"] * data["y"]
data["X^2"] = data["X"]**2
data["Y^2"] = data["Y"]**2
```

```
data
```

	x	y	X	Y	X*Y	X^2	Y^2
0	6.1101	17.59200	-2.0497	11.752865	-36.058322	4.201270	138.129834
1	5.5277	9.13020	-2.6321	3.291065	-24.031599	6.927950	10.831108
2	8.5186	13.66200	0.3588	7.822865	4.901926	0.128737	61.197216
3	7.0032	11.85400	-1.1566	6.014865	-13.710336	1.337724	36.178600

```
# we need to have the summation of all these
summation_x_y = data["X*Y"].sum()
summation_x_x = data["X^2"].sum()
summation_y_y = data["Y^2"].sum()


# correlation
correlation = summation_x_y/(summation_x_x*summation_y_y)**0.5
```

95	13.3940	9.05510	5.2342	3.215965	47.396204	27.396850	10.342431

```
correlation
```

```
0.837873232526341
```

```
#  we need to find out the Standard deviation of x and y
import statistics
std_dev_x = statistics.stdev(data["x"])
std_dev_y = statistics.stdev(data["y"])
print(std_dev_x,std_dev_y)
```

```
3.8698835278823314 5.510262255231544
```

```
# we need to find the slope
m = correlation * (std_dev_y/std_dev_x)


c = y_mean - m*x_mean


c
```

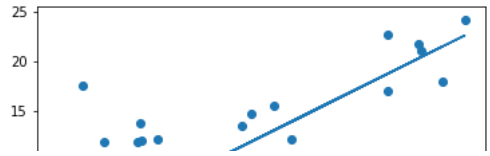
```
-3.8957808783118555
```

```
data["y_prediction"] = m * data["x"] + c
data
```

	x	y	X	Y	X*Y	X^2	Y^2	y_prediction
0	6.1101	17.59200	-2.0497	11.752865	-36.058322	4.201270	138.129834	3.393774
1	5.5277	9.13020	-2.6321	3.291065	-24.031599	6.927950	10.831108	2.698951
2	8.5186	13.66200	0.3588	7.822865	4.901926	0.128737	61.197216	6.267196
3	7.0032	11.85400	-1.1566	6.014865	-13.710336	1.337724	36.178600	4.459272
4	5.8598	6.82330	-2.3000	0.984165	-15.693590	5.290000	0.968581	3.095158
...	...	...	...	...	...	...	...	...
92	5.8707	7.20290	-2.2891	1.363765	-16.488158	5.239979	1.859855	3.108162
93	5.3054	1.98690	-2.8544	-3.852235	-5.671407	8.147599	14.839715	2.433740
94	8.2934	0.14454	0.1336	-5.694595	0.019311	0.017849	32.428413	5.998524
95	13.3940	9.05510	5.2342	3.215965	47.396204	27.396850	10.342431	12.083712
96	5.4369	0.61705	-2.7229	-5.222085	-1.680165	7.414184	27.270172	2.590624

97 rows × 8 columns

```
plot1 = plt.scatter(data["x"],data["y"])
plot2 = plt.plot(data["x"],data["y_prediction"])
```




```
# we need to find sse , ssr , sst and r2
ssr = ((data["y_prediction"] - y_mean)**2).sum()
ssr
```

```
2046.3146047180417
```

```
sse = ((data["y"]- data["y_prediction"])**2).sum()
sse
```

```
868.5324469391848
```

```
sst = sse + ssr
sst
```

```
2914.8470516572265
```

```
r2 = correlation**2
```